Nonlinear Dynamics and Systems Theory, **7** (1) (2007) 23–34

Inverse Determination of Model Parameters of Nonlinear Heat Conduction Problem Using Hybrid Genetic Algorithm

Shouju Li and Yingxi Liu

State Key Laboratory of Structural Analysis of Industrial Equipment, Dalian University of Technology, Dalian 116023, China

Received: June 17, 2005; Revised: October 16, 2006

Abstract: A new interpretation is proposed to solve the inverse heat conduction problem using hybrid genetic algorithm. In order to identify parameters of non-linear heat transfer efficiently and in a robust manner, the hybrid genetic algorithm, which combines genetic algorithm with simulated annealing and the elitist strategy, is presented for the identification of the material thermal parameters. The procedure is based on the minimization of an objective function which accounts for experimental data and the calculated response of the mathematical model. The performances of the proposed optimization algorithm were investigated with simulating data, and the effectiveness was consequently confirmed.

Keywords: *Inverse heat conduction problem; evolutionary algorithm; objective function; optimization algorithm; measurement noise.*

Mathematics Subject Classification (2000): 65N21.

1 Introduction

The accurate knowledge of the heat transfer coefficients is of importance in many engineering applications, including the cooling of continuously cast slabs and of electronic chips. In order to determine the heat transfer coefficients of materials, some identification methods have been developed for solving the problem [1]. For example, the sensitivity coefficient method was developed to solve multidimensional inverse heat conduction problems. The sensitivity coefficients are used directly to estimate the responses of the system considered under unit loading conditions. The finite-element discretization procedure is applied to evaluate the total response under all loading conditions. The conjugate gradient method is a powerful minimization technique, which can be applied to parameter and function estimations, as well as to linear and nonlinear inverse problems [2]. The conjugate gradient method with a suitable stopping criterion belongs to the class of iterative regularization techniques, where the number of iterations of the estimation procedure is determined so that stable solutions are obtained for the inverse problem [3]. The method consists in choosing a suitable direction of descent and a search step size along this direction at each of iteration for the minimization of the objective function.

The direct heat problem is concerned with the determination of the temperature field when the heat transfer coefficient, as well as the physical properties, initial condition and other quantities appearing in the boundary condition are known. Direct heat transfer problems can be mathematically classified as well-posed. The solution of a well-posed problem is required to satisfy the conditions of existence, uniqueness, and stability with respect to the input data. The inverse heat transfer problem is usually ill-posed. An ill-posed problem is characterized by the non-uniqueness and instability of solution. The regularization technique has been employed to overcome the ill-posedness of inverse heat transfer problems. Several such techniques have been introduced in the literature [4]. Most of the literature, however, uses a gradient-based optimization method and the solution often vibrates or diverges, depending upon the initial search point, since the model and the measurement errors can make the objective function complex [5]. There are numerous nonlinear optimization algorithms that could be employed in this problem. However, many nonlinear optimization techniques suffer from at least one of the tow shortcomings: either they are overly computationally intensive, or they tend to get trapped in local optima. One of the approaches used to overcome this problem is to use a robust optimization method and computational intelligences have been most successfully used to find the parameter set in a stable manner [6]. Genetic algorithm is effective nonlinear optimization techniques. It is based on the general approach apparent in nature by which species of organisms adapt, change, and improve. It is different from traditional optimization techniques in several ways. The genetic algorithm has been widely used in the identification, short-term load forecasting, the design optimization, dynamic channel assignment, the parameter identification of inelastic constitutive models [7, 8, 9, 10].The main purpose of the paper is to present a procedure for determining the thermal parameters in a robust manner.

2 Direct Problems for Heat Transfer

The partial differential equation governing the steady-state temperature distribution in a two-dimensional region described by the Cartesian coordinates, x and$y$, takes the form

$$\frac{\partial}{\partial x}(k_x\frac{\partial T}{\partial x})+\frac{\partial}{\partial y}(k_y\frac{\partial T}{\partial y})=0, \tag{1}$$

where T is the temperature, and k_x and k_y are the thermal conductivities in the x and y directions, respectively. The physical parameters, k_x and k_y, can be treated to be temperature dependent. The following three kinds of boundary conditions occur in the direct heat transfer problems, prescribed temperature (Dirichlet type), prescribed heat flux (Neumann type), and prescribed heat transfer coefficient (mixed or Robin type).

$$\begin{aligned}&T(x,y)=T_0, \qquad (x,y)\in\Omega_1,\\&k_x(\partial T/\partial x)n_x+k_y(\partial T/\partial y)n_y=q_0, \qquad (x,y)\in\Omega_2,\\&k_x(\partial T/\partial x)n_x+k_y(\partial T/\partial y)n_y=h(T_a-T), \qquad (x,y)\in\Omega_3,\end{aligned} \tag{2}$$

where T_0 represents the value on the surface or boundary Ω_1, q_0 is a heat flux vector on subsurfaceΩ_2, h is the convective heat transfer coefficient on subsurface Ω_3, n_x and n_y are the direction cosines of the outward drawn normal to the boundary, T_a is the surrounding temperature.

3 Inverse Problem for Heat Transfer and solution Approach with Hybrid Genetic Algorithm

3.1 Definition of inverse problem for parameter identification

For the inverse problem, the heat transfer coefficient is regarded as unknown. The parameter identification problem can be formulated to find the model parameters by adjusting identified parameter vector $\boldsymbol{m}$ until the measured data match the corresponding data computed from the parameter set in a least-squares fashion. The objective function is defined as follows [3]

$$J(m) = [h_m - h_c(m)]^T w[h_m - h_c(m)], \tag{3}$$

where h_m is the measured temperature vector; h_c is the computing temperature vector, which is related to the identified parameter vector $\boldsymbol{m}$, $\boldsymbol{w}$ is weighting matrix in order to take into account the different observed equipments for the temperature measurements. This objective function clearly depends on the measured data and the parameters of model. The objective function can become complex, such as non-convex, or even multi-modal if errors contained in the model equation or /and errors in the measurement data are large.

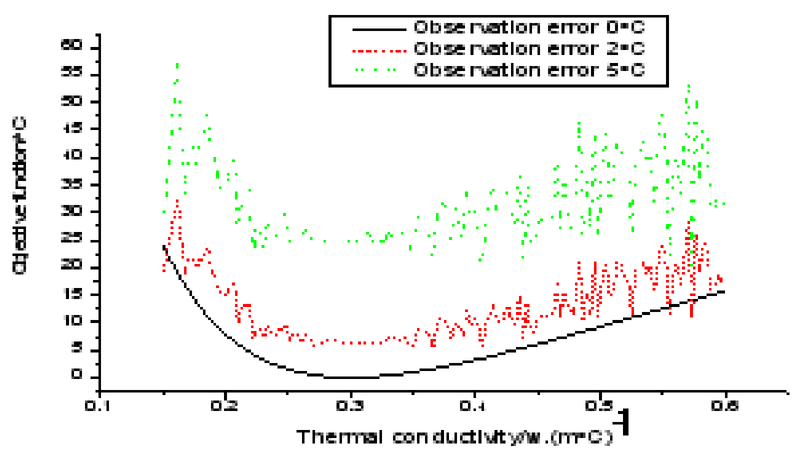

Figure 3.1: Objective functions of the different measurement errors.

When taking account of measurement errors, the objective function will have many local minima for the inverse heat conduction problem, as shown in Figure 3.1. When random measurement error is 2 $^\circ$C, the objective function has 35 local minima. When random measurement error is $5\,^\circ$C, the objective function has 38 local minima. In such a case, the solution may vibrate or diverge when conventional gradient-based optimization methods are used, which gives rise to the necessary for a robust optimization method such that a stable convergence is always achieved. The solution of the inverse problem consists in obtaining a minimum of an objective function which is defined taking into account the mathematical structure of the material model and asset of experimental data. This generally results in a non-linear programming constrained problem of the form [2]:

$$\min\{J(m, h_m)\ ,\quad m \in R^{\ P}, h_m \in R^M; g^j < 0\}, \tag{4}$$

where $\boldsymbol{m}$ belongs to the space of admissible parameters R^P, h_m belongs to the space R^M, g^j are inequality constraints, which define the feasible domain

$$D = \{m \in R^P, g^j < 0\}. \tag{5}$$

The constraints can represent physical links between the primary physical variables and the model parameters, information concerning the values of parameters and conditions to guarantee that all mathematical functions involved can be defined and calculated. In the optimization process, the difference between the experimental result and the theoretical prediction is measured by a norm value J here referred to as the Euclidean norm. The Euclidean norms of the tests form an objective function $J(m)$ which then gives a scalar measure of the error between the experimental observations and the model predictions. From mathematical point of view, the optimization problem involves the minimization of the objective function [4]:

$$J(m) \to \min. \tag{6}$$

The bound constraints:

$$m_l < m < m_u, \tag{7}$$

where m_l and m_u are, respectively, the lower and upper bounds of m. Traditional mathematical optimization methods that have been used include dynamic programming, conjugate-gradient, random search, and simplex optimization.

3.2 Continuous Evolutionary Algorithm and Its Improvements

Genetic algorithm (GA) is a search method based on Darwin's theory of evolution and survival of the fittest [11]. Based on the concept of genetics, GA simulates the evolutionary process numerically. Genetic algorithms strongly differ in conception from other search methods, including traditional optimization methods and other stochastic search methods. The basic difference is that while other methods always process single points in the search space, genetic algorithms maintain a population of potential solutions. Genetic algorithms constitute a class of search methods especially suited for solving complex optimization problems. Search algorithms in general consist of systematically walking through the search space of possible solutions until an acceptable solution is found. Genetic algorithms transpose the notions of natural evolution to the world of computers, and imitate natural evolution. They were initially introduced by John Holland for explaining the adaptive processes of natural systems and for creating new artificial systems that work on similar bases [12]. In Nature new organisms adapted to their environment develop through evolution. Genetic algorithms evolve solutions to the given problem in a similar way. The main contents of genetic algorithm include the evaluation of fitting function, selection operation, crossover operation, mutation and elitist strategy.

The probability of survival of any individual is determined by its fitness: through evolution the fitter individuals overtake the less fit ones. In order to evolve good solutions, the fitness assigned to a solution must directly reflect its 'goodness', i.e. the fitness function must indicate how well a solution fulfills the requirements of the given problem. The evaluation of the fitness can be conducted with a linear scaling, where the fitness of each individual is calculated as the worst individual of the population subtracted from its objective function value. Fitness assignment can be performed in several different ways: We define a fitness function and incorporate it in the genetic algorithm. When evaluating any individual, this fitness function is computed for the individual.

$$f_j = \max\{J_j/j = 1, 2, , , S\} - J_j, \tag{8}$$

where f_j is the fitting function of j-th individual; S is the population size; J_j is the objective function of j-th individual. Selection, also called reproduction, is simply the copying of quality solution in proportion to their effectiveness. Here, since the goal is to minimize the objective function, several copies of candidate solutions with small objective functions are made; solutions with large objective functions tend not to be replicated. The intrinsic principle of the genetic algorithm is Darwin's natural selection principle. Selection is the impetus of the genetic algorithm, by which, the superior individual are selected into the next generation while the inferior ones are washout. A part of the new population can be created by simply copying without change selected individuals from the present population.

One of the most commonly used is the roulette wheel selection, where individuals are extracted in probability following a Monte Carlo procedure. The extraction probability of each individual is proportional to its fitness as a ratio to the average fitness of all the individuals. In the selection process, the reproduction probabilities of individuals are given by their relative fitness:

$$pro_j = f_j / \sum_{j=1}^{s} f_j, \tag{9}$$

where pro_j is the reproduction probability of the j-th individual.

Recombination, also called crossover, is a process by which information contained in two candidate solutions is combined. In the recombination, each individual is first paired with an individual at random. New individuals are generally created as offspring of two parents (as such, crossover being a binary operator). One or more so-called crossover points are selected (usually at random) within the chromosome of each parent, at the same place in each. The parts delimited by the crossover points are then interchanged between the parents. The individuals resulting in this way are the offspring. Beyond one point and multiple point crossover there exist more sophisticated crossover types. Let a pair of present individuals be given by $[m_\alpha^t, m_\beta^t]$. a new pair $[m_\alpha^{t+1}, m_\beta^{t+1}]$ is then created in terms of a phenomenological recombination formula[8]:

$$m_\alpha^{t+1} = (1-\mu)\, m_\alpha^t + \mu \cdot m_\beta^t, \tag{10}$$

$$m_\beta^{t+1} = (1-\mu)\, m_\beta^t + \mu \cdot m_\alpha^t, \tag{11}$$

where μ is a random number changing from 0 to 1.

A new individual is created by making modifications to one selected individual. The modifications can consist of changing one or more values in the representation or in adding/deleting parts of the representation. In genetic algorithms mutation is a source of variability, and is applied in addition to crossover and reproduction. Mutation is a process by which vectors resulting from selection and recombination are perturbed. The mutation is conducted with only a small probability by definition. An individual, after this mutation m_i^{t+1}, is described as

$$m_i^{t+1} = rand\left\{m_l, m_u\right\}, \tag{12}$$

where $rand\{.\}$ represents the random selection from the reasonable solution domains. At different stages of evolution, one may use different mutation operators. At the beginning mutation operators resulting in bigger jumps in the search space might be preferred. Later on, when the solution is close by, a mutation operator leading to slighter shifts in the search space could be favored. However, the above mutation operation is a random one with no clear aim.

Simulated annealing is another important algorithm which is powerful in optimization and high-order problems [13]. It uses random processes to help guide the form of its search for minimal energy states. In an annealing process a melt, initially at high temperature and disordered, is slowly cooled so that the system at any time is approximately in thermodynamic equilibrium. As cooling proceeds, the system becomes more ordered and approaches a "frozen" ground state at T=0. The paper provides a mutation method based on the simulating annealing algorithm, which makes the average fitness of the population tend to be optimized. Firstly, we define a neighborhood structure, then select a new solution in the neighborhood structure of the intermediate solution, that is to say, getting a new solution by cause a disturb on the old one [14]

$$m_{new} = m_{old} + \Delta m, \tag{13}$$

where m_{new} and m_{old} represents a new solution and an old solution, respectively; $\Delta m$is a random disturb. Then, reject or accept the new solution according the Metropolis rule, the probability of accepting the new generated solution is expressed as the follows [15]:

$$p_{new} = \left\{ \begin{array}{ll} 1 & J_{old} \geq J_{new} \\ \exp[-\Delta J/T_k] & J_{old} < J_{new} \end{array} \right\}, \tag{14}$$

where J_{new} and J_{old} are the objective functions of the new solution and the old solution; p_{new} is the probability of accepting the new generated solution; $\Delta J$is the increasement of the objective function, $\Delta J = J_{new} - J_{old}$, T_k is the annealing temperature, which tends to be drooped during the evolutional process. The probability of rejecting the new solution is:

$$P_{old} = 1 - p_{new}, \tag{15}$$

where p_{old} is the probability of rejecting the new solution. One feature that is currently missing in this selection procedure is that it does not guarantee the best individual always survives into the next generation, particularly when many individuals have fitness close to that of the best individual. The elitist strategy, where the best individual is always survived into the next generation on behalf of the worst individual, can compensate for some disadvantages of missing the best individual in selection operation or mutation operation. With the elitist strategy, the best individual always moves in a descent direction, thereby a stable convergence is obtained. The gradient search algorithm adopted in genetic algorithm is the most popular quasi-Newton method with the BFGS algorithm. The individual after the recombination is formulated as follows [8]

$$m_{new} = \left\{ \begin{array}{ll} -A^{-1}\nabla f\left(m_{old}\right) & if \quad f\left(m_{new}\right) > f\left(m_{old}\right) \\ m_{old} & otherwise \end{array} \right\}, \tag{16}$$

where A is a well-known positive-define matrix used on behalf Hessian matrix. The main steps for parameter identification using genetic algorithm are shown in follows:

Step 1: Choose the size of population, the crossover probability, mutation probability, and stopping criterion.

Step 2: Determine the identified parameter domains according to prior information

Step 3: Randomly generate an initial population of candidate solutions.

Step 4: Define a fitness function to measure the performance of an individual in the problem domain.

Step 5: Compute model responses with given model parameters by using Newton iteration method.

Step 6: Calculate the fitness of each individual based on observed dada and model responses computed by Newton iteration method.

Step 7: Execute recombination operation by using continuous floating codes.

Step 8: Create new individuals by mutation operation based on simulated annealing.

Step 9: Execute select operation according to the roulette wheel selection.

Step 10: Perform elitist strategy in order to keep current best individual from missing and accelerate convergence speed of inverse problem.

Step 11: Replace the initial(parent) population with the new (offspring) population.

Step 12: Execute stopping criterion. If stopping criterion can not be reached, then, go to Step 5; otherwise, the inversion computation stops and best solution is recorded as the solution of the inverse problem.

4 Numerical Examples for Identification of Model Parameters in Heat Transfer

In order to demonstrate the accuracy of the proposed algorithm with nonlinear problem, a heat transfer problem with nonlinear material properties is considered in this example. An annular cylinder is subjected to a constant prescribed heat flux, q_0, on the outer surface, and a temperature T_0, prescribed on the inner surface, and shown in Figure 4.1.

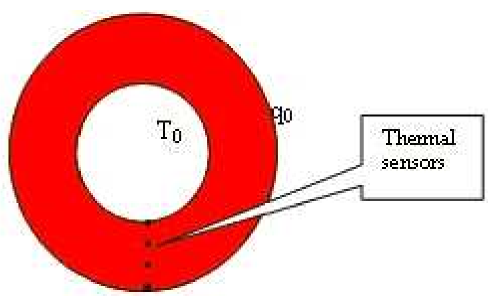

Figure 4.1: An annular cylinder and sensor locations.

The governing equation of heat transfer can be expressed as

$$\frac{1}{r}\frac{d}{dr}\left[k\left(T\right)r\frac{dT}{dr}\right]=0, \tag{17}$$

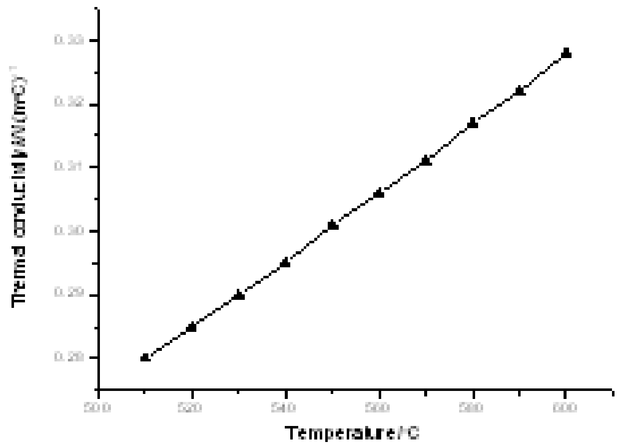

Figure 4.2: Change of thermal conductivity versus temperatures.

where $k(T)$ is a quadratic function of temperature; r is radial coordinate. And k can be expressed as follows

$$k(T) = k_0 + k_1 T + k_2 T^2, \tag{18}$$

where k_0, k_1, and k_2 are unknown material thermal parameters. The thermal conductivity is depended on the temperatures as shown in Figure 4.2. The change of thermal conductivity versus temperatures is shown in Figure 4.2.

The analytical solution of Equation (17) can be expressed as follows:

$$k_0 (T - T_0) + k_1 (T - T_0) + k_2 (T - T_0)^2 = q_0 R_0 \ln (r/R_i), \tag{19}$$

where R_i is the inner radius, R_i =1.0 m; R_o is the outer radius of the annular cylinder, R_o =2.0 m; q_0 is a constant prescribed heat flux, q_o=10 W/m^2. The above algebraic equation can be solved numerically by using Newton iteration algorithm. Suppose the material thermal parameters are known, the simulated measured temperature values are shown in Figure 4.3.

Contrasting with the direct analysis, the inverse heat transfer analysis is ill-conditioned; the latter predicts the surface temperature or heat flux across the surface using temperature measured at certain discrete points inside the domain considered. Figure 4.4 is the objective function value versus number of iterations for the optimal individual. Figure 4.4 shows that the hybrid genetic algorithm can make searching more accurate and faster near global minima on the error surface.

Table 4.1 shows the comparison of the identified thermal conductivity by using different inverse methods without measurement errors. Identified values (A) and (B) shown in Table 4.1 represent for the identified thermal conductivity by using hybrid genetic algorithm and classical genetic algorithm, respectively. The hybrid genetic algorithm has higher identification precision than classical genetic algorithm.

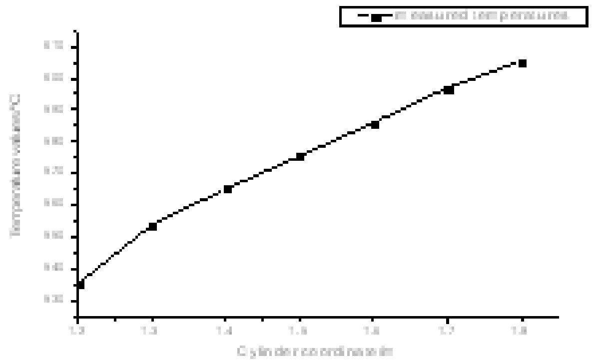

Figure 4.3: Simulated measurement temperatures in sensor.

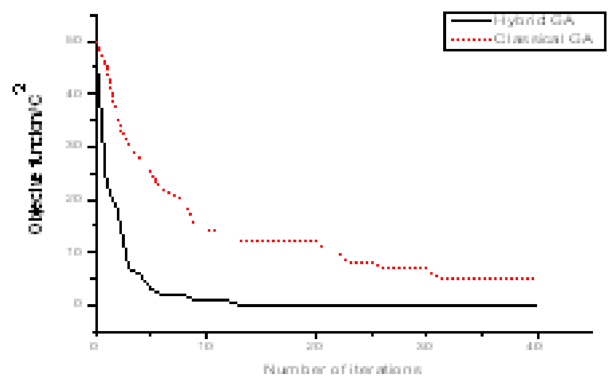

Figure 4.4: Objective function value versus number of iterations for the optimal individual.

Parameters	$k_0/W{\cdot}(m\ ^\circ C)^{-1}$	$k_1/10^{-3}$ $*Wm{\cdot}(m\ ^\circ C)^{-2}$	$k_2/10^{-6}$ $*Wm^2{\cdot}(m\ ^\circ C)^{-3}$
Theoretical values	0.100	0.198	0.303
Identified values(A)	0.101	0.199	0.302
Identified values(B)	0.112	0.203	0.305

Table 4.1: Comparison of the identified thermal conductivity by using different inverse methods without measurement errors.

Often a small measurement error can dramatically change the boundary values predicted. The temperature measurement errors can come from many sources and can be caused by calibration errors, presence of the sensor, conduction and convection losses of the sensor, or signal analysis. Also, these can be further determined by errors from measurements of time, sensor location, or dimension of the domain considered. Several studies simulated measurement noise numerically by superimposing a random noise with zero mean and a specified variance on measured data. The proportional error, in which the uncertainty for each measured data is proportional to its own value, is used to study the influence of measurement noise on the identified thermal conductivity. These measurements were generated from the solution of the direct problem. The measurements containing random errors were obtained by adding an error term to the errorless measurements resulting from the solution of the direct problem; that is:

$$T_m = T_{exa} + \xi\sigma, \tag{20}$$

where T_{exa} are the errorless measurements; ξ is a random variable with normal distribution, zero mean, and unitary standard deviation; σ is the standard deviation of the measurement errors, which is supposed constant; and T_m are the measurements containing random errors. Three different levels of measurement errors are considered here. Table 4.2 shows the influences of measurement noises on the identified thermal conductivity.

Parameters	$k_0/W{\cdot}(m\,^\circ C)^{-1}$	$k_1/10^{-3}$ $*Wm{\cdot}(m\ ^\circ C)^{-2}$	$k_2/10^{-6}$ $*Wm^2{\cdot}(m\,^\circ C)^{-3}$
Theoretical values	0.100	0.198	0.303
Identified values(C)	0.103	0.202	0.310
Identified values(D)	0.107	0.195	0.322
Identified values(E)	0.984	0.210	0.298

Table 4.2: Influences of measurement noises on the identified thermal conductivity.

Identified values (C), (D) and (E) shown in Table 4.2 represent for the identified thermal conductivity when the measurement error of temperature is $1.0\,^\circ$C, $2.0\,^\circ$C and $5.0\,^\circ$C, respectively. The proposed scheme accurately identifies the thermal conductivity of nonlinear heat conduction problem for a large random error. As the random error becomes larger, detailed information on the thermal parameters is lost slightly, and the identification precision decreases.

5 Conclusion

The proposed identification scheme is based on the minimization of the least-squared errors between measured and calculated temperatures at observation points. The evolutionary algorithm is employed to overcome numerical difficulties caused by the ill-posedness of inverse problems. The validity and effectiveness of the proposed method are demonstrated by the examples. In order to demonstrate the stability of proposed method to measurement errors, the Monte-Carlo simulation is performed of various amplitudes of random errors in the example. And numerical computational results show the insensitivity of the proposed algorithm to measurement errors. Simulated temperature measurements of several sensors located inside the material were utilized for the estimation of the spatial variations of the heat transfer coefficient. Different cases are examined here, involving different numbers of sensors, levels of measurement error. It is shown that the proposed method converges very fast in a robust manner, and is not only simple and flexible, but also versatile and accurate. With considering measurement errors for 5%, the inversion method can identify the thermal conductivity within 1%, as compared with theoretical values.

Acknowledgements

This research is funded by the National Natural Science Foundation of China (No. 10472025).

References

[1] Tseng, A. A. Direct sensitivity coefficient method for solving two-dimensional inverse heat conduction problems by finite-element scheme. *Numerical Heat Transfer* **27**(2) (1995) 291–307.

[2] Marcelo, J. Comparison of different versions of the conjugate gradient method of function estimation. *Numerical Heat Transfer* **36**(2) (1999) 229–249.

[3] Lee, H. S. A new spatial regularization scheme for the identification of the geometric shape of an inclusion in a finite body. *Int. J. for Numerical Methods in Engineering* **46**(7) (1999) 973–992.

[4] Tervola, P. A method to determine the thermal conductivity from measured temperature profiles. *Int. J. Heat Mass Transfer* **32**(8) (1989) 1425–1431.

[5] Lin, J Y. Numerical estimation of thermal conductivity from boundary temperature measurements. *Numerical Heat Transfer* **32**(2) (1997) 187–203.

[6] Garcia, S. Use of genetic algorithms in thermal property estimation: simultaneous estimation of thermal properties. *Numerical Heat Transfer* **33**(2) (1998) 149–168.

[7] Yue, W. Y., Miyazaki, K., and Deng, X. T. Optimal channel assignment in wireless communication networks with distance and frequency interferences. *Computer Communications* **27**(16) (2004) 1661–1669.

[8] Furukawa, T. An automated system for simulation and parameter identification of inelastic constitute models. *Comput. Methods Appl. Mech. Eng.* **191**(6) (2002) 2235–2260.

[9] Coats, L. Evolutionary algorithms approach to the solution of mixed integer non-linear programming problems. *Computers and Chemical Engineering* **25**(2-3) (2001) 257–266.

[10] Sahab, M. G. A hybrid genetic algorithm for reinforced concrete flat slab buildings. *Computers and Structures* **83**(8-9) (2005) 551–559.

[11] Kustrin, K. Basic Concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research. *J. of Pharmaceutical and Biomedical Analysis* **22**(5) (2000) 717–727.

[12] Abbass, H. A. An evolutionary artificial neural networks approach for breast cancer diagnosis. *Artificial Intelligence in Medicine* **25**(3) (2002) 265–281.

[13] Alkhanmis, T. M. Simulated annealing for discrete optimization with estimation. *European Journal of Operational Research* **116**(3) (1999) 530–544.

[14] Jeong, I. K. Adaptive simulated annealing genetic algorithm for system identification. *Engineering Applications of Artificial Intelligence* **9**(5) (1996) 523–532.

[15] Chen, T. Y. Efficiency improvement of simulated annealing in optimal structural designs. *Advances in Engineering Software* **33**(7–10) (2002) 675–680.